7. Lineare Optimierung

7.1 Problemstellung

Ein Problem der linearen Optimierung – auch lineares Programm genannt – bezeichnet die Aufgabe, eine lineare Zielfunktion

$$z = f(x) = c^T x \quad , \quad x \in \mathbb{R}^n$$

unter linearen Gleichungs-/Ungleichungs=nebenbedingungen

$$a_i^T x = b_i \quad \text{oder} \quad a_i^T x \leq b_i$$

zu minimieren.

In Normalform (kanonischer Form) lautet die Problemstellung:

Problem (7.1) (Normalform)

Bestimme $x \in \mathbb{R}^n$ mit $z = f(x) = c^T x$ minimal unter den Nebenbedingungen

$$Ax = b \, , \quad (A \in \mathbb{R}^{(m,n)} \, , \; m < n) \, ,$$

$$x \geq 0 \, .$$

Zusatzvoraussetzungen (7.2)

ⓐ Rang (A) = m (maximal)

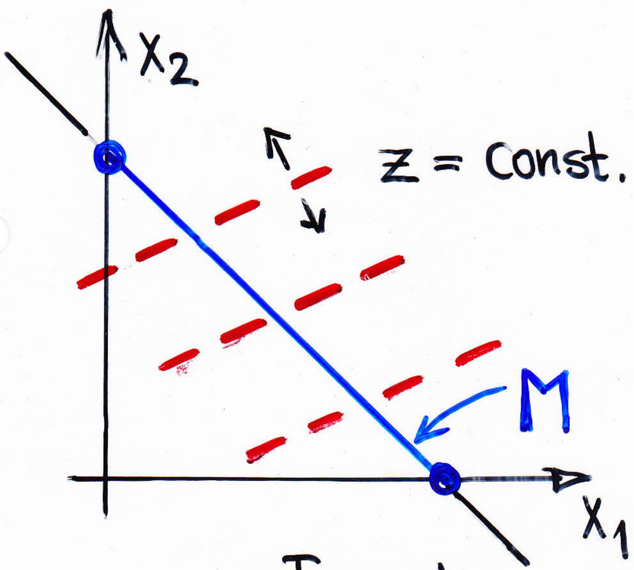
ⓑ Die Menge der zulässigen Punkte

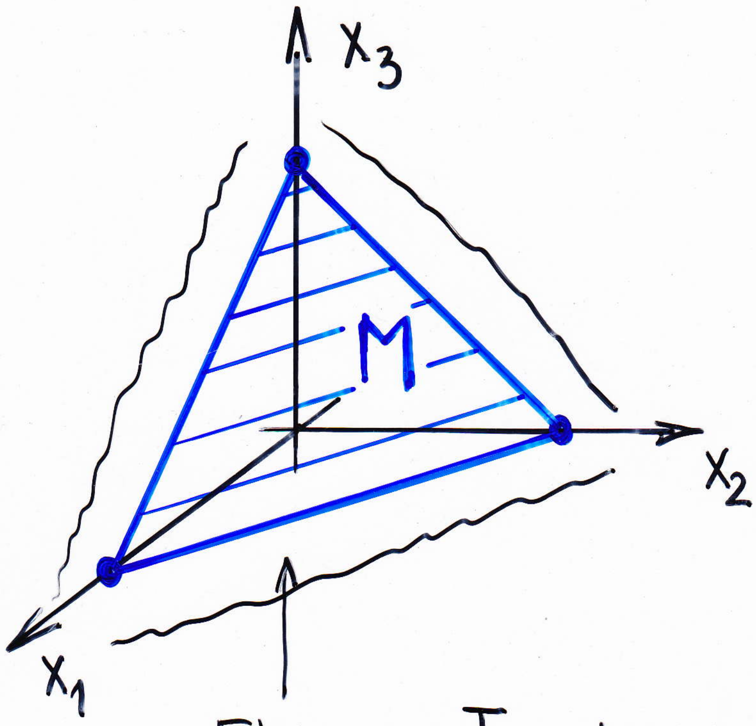
$$M := \{x \in \mathbb{R}^n \mid Ax = b \wedge x \geq 0\}$$

ist nichtleer.

Graphische Darstellung:

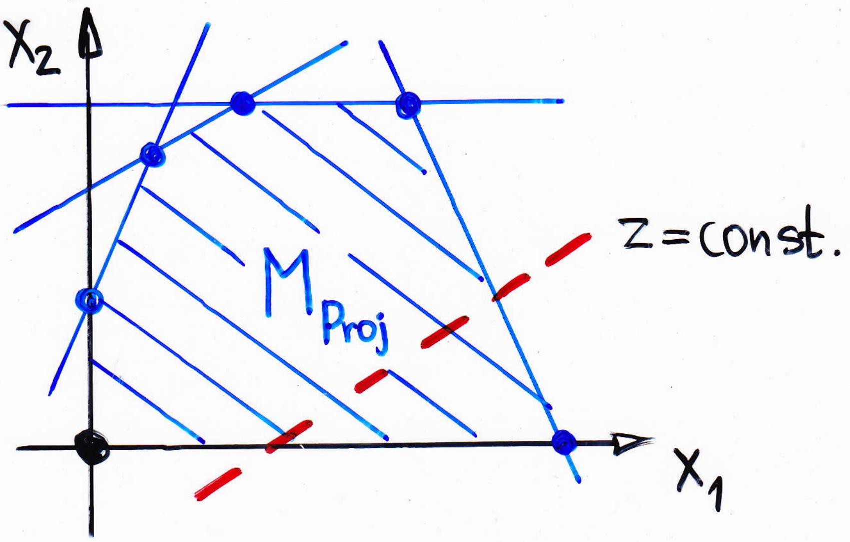
A: Gleichungen





$(n = 2/3, \quad m = 1)$

B: Ungleichungen



(i) Eine Ungleichung
$a_1 x_1 + a_2 x_2 \leq b$ (im $\mathbb{R}^2$) bzw.
$a^T x \leq b$ (im $\mathbb{R}^n$) beschreibt eine
Halbebene bzw. einen Halbraum.
Damit bildet die "projizierte" zulässige
Menge

$$M_{proj} := \{ \tilde{x} \in \mathbb{R}^{n-m} \mid \tilde{A}\tilde{x} \leq b, \ \tilde{x} \geq 0 \}$$

ein Konvexes Polyeder.

(ii) Die Höhenlinien der Zielfunktion

$$z = c^T x = z_0 = const.$$

bilden eine Schar paralleler Hyperebenen.

Beispiel (7.3) "Schuhfabrik"

	Modell 1	Modell 2	maximal
Herstellungszeit [h]	20	10	8000
Maschinen [h]	4	5	2000
Leder [dm^2]	6	15	4500
Gewinn [€]	16	32	?

$x_{1,2}$: Anzahl der Modelle 1/2

Aufgabe:

Maximiere den Gewinn $16x_1 + 32x_2$

unter den Nebenbedingungen:

$$\begin{aligned} 20x_1 + 10x_2 &\leq 8000 \\ 4x_1 + 5x_2 &\leq 2000 \\ 6x_1 + 15x_2 &\leq 4500 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Zur Herstellung der Normalform werden **Schlupfvariable** x_3, x_4, x_5 eingeführt.

Damit lautet die Aufgabe:

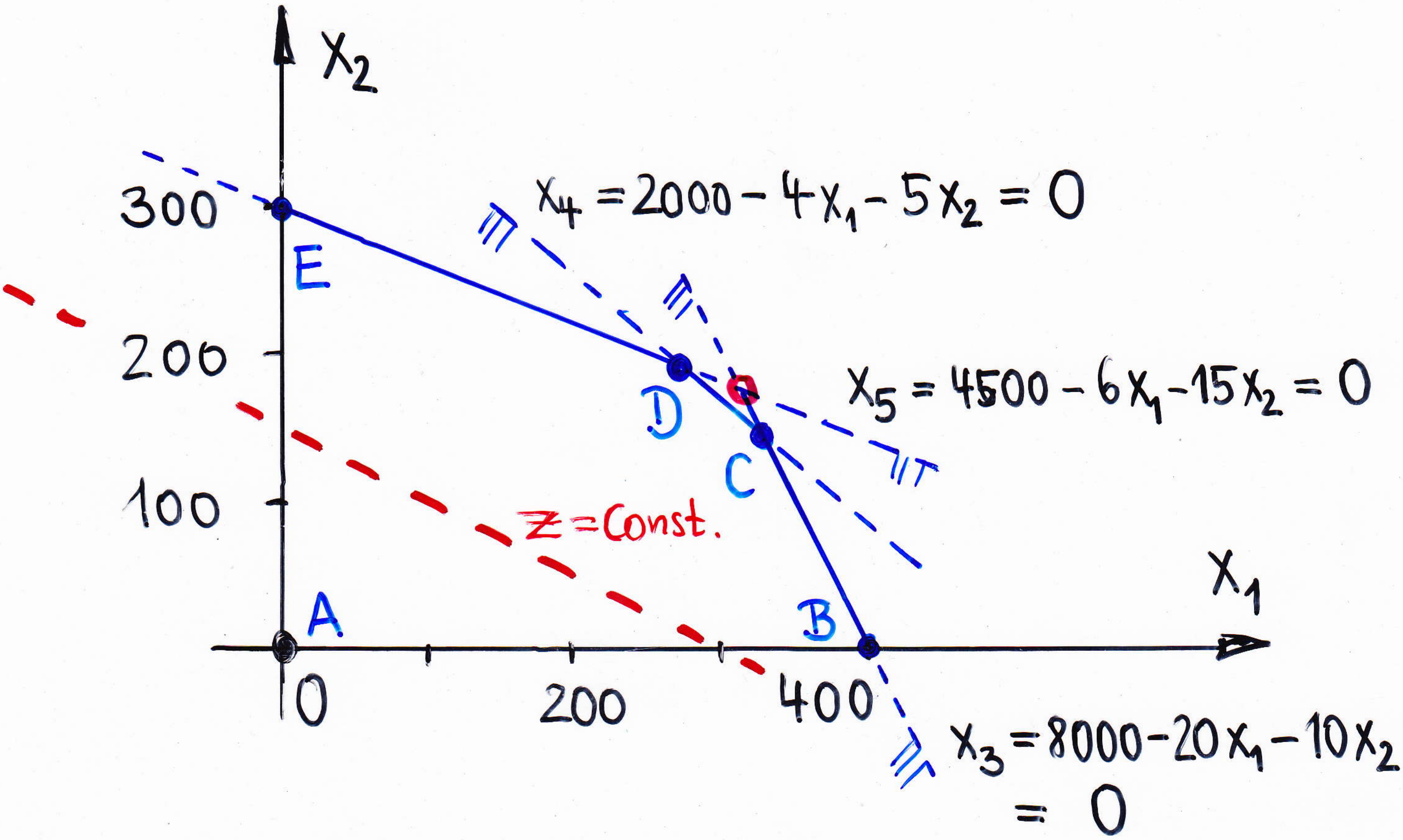
Minimiere

$$z = -16x_1 - 32x_2 \; ; \quad c^T := (-16, -32, 0, 0, 0)$$

unter den Nebenbedingungen:

$$\begin{pmatrix} 20 & 10 & 1 & 0 & 0 \\ 4 & 5 & 0 & 1 & 0 \\ 6 & 15 & 0 & 0 & 1 \end{pmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_5 \end{bmatrix} = \begin{pmatrix} 8000 \\ 2000 \\ 4500 \end{pmatrix}$$

$$x_1, x_2, \dots x_5 \geq 0$$



Man erkennt anhand des Beispiels:

- Ist M nichtleer und beschränkt, so besitzt das lineare Programm (wenigstens) eine Lösung x^*, die in einer Ecke von M_{proj} liegt.
- Man erhält die Ecken $(A, \dots, E)$, indem man zwei der Variablen $x_1, \dots, x_5$ Null setzt und die anderen aus $Ax = b$ berechnet.

  $A: x_1 = x_2 = 0$, $B: x_2 = x_3 = 0$,

  $C: x_3 = x_4 = 0$, ...

  Beachte, dass nicht alle möglichen "Ecken" zulässig sind, z.B. führt $x_3 = x_5 = 0$ auf eine nicht zulässige "Ecke" $\frac{1}{2}(625, 350)^T$.

Definition (7.4)

a) Für $x \in M$ heißt $I(x) := \{ j : x_j > 0 \}$ die Menge der inneren Indizes.

b) $x \in M$ heißt eine Basislösung zur Indexmenge J_B, falls gelten:

$$J_B \subset \{1, .., n\}, \quad \# J_B = m,$$

$$I(x) \subset J_B, \quad (a^j)_{j \in J_B} \text{ Basis des } \mathbb{R}^m$$

(Dabei: $A = (a^1, \dots, a^n)$, $a^j \in \mathbb{R}^m$, $m < n$.)

Die Indizes $j \in J_B$ heißen Basisindizes oder Basisvariable, die $j \in J_N := \{1, ..., n\} \setminus J_B$ heißen Nichtbasisvariable.

Eine Basislösung heißt entartet, falls $I(x) \subsetneq J_B$.

Satz (7.5) (Fundamentalsatz)

Ist ein lineares Programm (7.1) lösbar, so existiert auch eine Lösung x^*, die zugleich eine (zulässige) Basislösung ist.

Beweis :

Sei $x \in \mathbb{R}^n$ eine Lösung von (7.1).

1. Fall : Sind die $(a^j)_{j \in I(x)}$ linear unabhängig, so lassen sie sich wegen $\text{Rang}(A) = m$ zu einer Basis $(a^j)_{j \in J_B}$, $I(x) \subset J_B$ ergänzen ($\rightarrow$ lineare Algebra). Damit ist x Basislösung zur Indexmenge J_B.

2. Fall : $(a^j)_{j \in I(x)}$ linear abhängig!

Wir ersetzen x durch eine andere Lösung $\tilde{x}$ von (7.1) mit $I(\tilde{x}) \subsetneqq I(x)$.
Nach endlich vielen solcher Ersetzungsschritte ist dann Fall 1 erreicht!

Nach Voraus. ex. $y \in \mathbb{R}^n \setminus \{0\}$ mit $I(y) \subset I(x)$ und $Ay = 0$.
Bestimme $i_0 \in I(x)$ mit

$$\mu := \min \left\{ \left| \frac{x_i}{y_i} \right| : i \in I(x),\ y_i \neq 0 \right\} = \left| \frac{x_{i_0}}{y_{i_0}} \right| > 0$$

und setze $\tilde{x}^{\pm} := x \pm \mu y$.

Damit folgt:

- $A\tilde{x}^{\pm} = Ax \pm \mu Ay = Ax = b$,
- $\tilde{x}^{\pm} \geq 0$ (nach Def. von μ),
- Da beide Punkte $\tilde{x}^{\pm}$ zulässig sind und $f(\tilde{x}^{\pm}) = c^T x \pm \mu (c^T y)$ folgt (x optimal!) $c^T y = 0$

  $\Rightarrow$ Beide Punkte $\tilde{x}^{\pm}$ sind optimal!
- Wählt man das Vorzeichen $-\operatorname{sgn}\left(\frac{x_{i_0}}{y_{i_0}}\right)$, so folgt:

  $$\tilde{x}_{i_0} = x_{i_0} - \operatorname{sgn}\left(\frac{x_{i_0}}{y_{i_0}}\right) \cdot \left|\frac{x_{i_0}}{y_{i_0}}\right| \cdot y_{i_0} = 0.$$

  Damit ist $\tilde{x}$ eine weitere Lösung mit $I(\tilde{x}) \subsetneqq I(x)$ ∎

Bemerkung (7.6)

Ist in obigem Beweis x lediglich zulässig ($Ax = b$, $x \geq 0$), so lässt sich eine Basislösung $\tilde{x}$ konstruieren mit $I(\tilde{x}) \subset I(x)$!

Satz (7.7) (Darstellungssatz)

Es bezeichne $\{v_\ell \mid \ell \in L\}$ die (endliche und nichtleere) Menge der Basislösungen. Jeder zulässige Punkt $x \in M$ besitzt eine Darstellung

$$x = \sum_{\ell \in L} \alpha_\ell v_\ell + d$$

mit $\alpha_\ell \geq 0$, $\sum \alpha_\ell = 1$, $d \geq 0$, $Ad = 0$.

Beweis: (vollst. Induktion über $p := \# I(x)$)

$p = 0$: $\Rightarrow$ $x = b = 0$. Damit ist x selbst Basislösung, die Darstellung also trivial erfüllt.

$0, 1, .., p-1 \Rightarrow p$: Sei x o.E.d.A. keine Ecke. Dann sind die $(a^j)_{j \in I(x)}$ linear abhängig

$$\Rightarrow \exists\, y \in \mathbb{R}^n \setminus \{0\} : I(y) \subset I(x) \wedge Ay = 0$$

Fall 1: y hat positive u. negative Komponenten

Setze:
$$\mu_1 := \min\left\{ \frac{x_i}{y_i} : i \in I(x),\ y_i > 0 \right\} > 0,$$

$$\mu_2 := \min\left\{-\frac{x_i}{y_i} : i \in I(x), y_i < 0\right\} > 0,$$

$$x^1 := x - \mu_1 y,$$

$$x^2 := x + \mu_2 y.$$

Wie in (7.5) folgt:

$$x^1, x^2 \in M, \quad I(x^1), I(x^2) < p.$$

Auf x^1, x^2 lässt sich also die Induktions=voraussetzung anwenden!

Schließlich ist

$$x = (1-\lambda)x^1 + \lambda x^2, \quad \lambda := \frac{\mu_1}{\mu_1 + \mu_2} \in]0,1[$$

und mit $x^i = \sum_{l \in L} \alpha_l^i v_l + d^i$, $i = 1,2$, wird

$$x = \sum_{l \in L} \underbrace{\left((1-\lambda)\alpha_l^1 + \lambda \alpha_l^2\right)}_{=: \alpha_l} v_l + \underbrace{\left((1-\lambda)d^1 + \lambda d^2\right)}_{=: d}$$

Dies zeigt die gewünschte Darstellung!

Fall 2: $y \geq 0$

Setze:

$$\mu_1 := \min\left\{\frac{x_i}{y_i} : i \in I(x), y_i > 0\right\} > 0,$$

$$x^1 := x - \mu_1 y .$$

Wieder folgt: $x^1 \in M$, $I(x^1) < p$, also nach Induktionsvorauss.:

$$x^1 = \sum_{\ell \in L} \alpha_\ell^1 v_\ell + d^1$$

$$\Rightarrow \quad x = \sum_{\ell \in L} \alpha_\ell^1 v_\ell + \underbrace{(d^1 + \mu_1 y)}_{=:d}$$

Fall 3: $y \leq 0$, genauso! ∎

Satz (7.8) (Existenzsatz)

Für ein lineares Programm (7.1), das die Vorauss. (7.2) erfüllt, gilt stets:

Entweder: $\inf\{f(x) : x \in M\} = -\infty$

oder: $\exists$ optimale Basislösung!

Beweis:

Fall 1: Es gibt ein $d \in \mathbb{R}^n$ mit:

$$d \geq 0 , \quad Ad = 0 , \quad c^T d < 0 .$$

Ist $x_0 \in M$, so folgt für $t \geq 0$:

$$x_0 + td \in M,$$

$$c^T x = c^T x_0 + t(c^T d) \to -\infty \quad (t \to \infty).$$

Fall 2:

$$\forall d \in \mathbb{R}^n : \left(d \geq 0 \wedge Ad = 0 \Rightarrow c^T d \geq 0 \right)$$

Nach (7.7) gilt für jedes $x \in M$ die Darstellung

$$x = \sum_{\ell \in L} \alpha_\ell v_\ell + d,$$

wo v_ℓ : Basislösung, $\alpha_\ell \geq 0$, $\sum \alpha_\ell = 1$, $d \geq 0$, $Ad = 0$ (also $c^T d \geq 0$).

$\Longrightarrow$

$$c^T x = \sum_{\ell \in L} \alpha_\ell (c^T v_\ell) + (c^T d)$$

$$\geq \min\{c^T v_\ell : \ell \in L\}$$

Damit ist unter den v_ℓ eine optimale Basislösung! ■

George Dantzig (1914 – 2005)

1947 : Simplex-Verfahren zur
Linearen Optimierung

Simplex Method

aus: Nocedal, Wright: Numerical Optimization, 1999

Dantzig's development of the simplex method in the late 1940s marks the start of the modern era in optimization. This method made it possible for economists to formulate large models and analyze them in a systematic and efficient way. Dantzig's discovery coincided with the development of the first digital computers, and the simplex method became one of the earliest important applications of this new and revolutionary technology. From those days to the present, computer implementations of the simplex method have been continually improved and refined. They have benefited particularly from interactions with numerical analysis, a branch of mathematics that also came into its own with the appearance of digital computers, and have now reached a high level of sophistication.

Today, linear programming and the simplex method continue to hold sway as the most widely used of all optimization tools. Since 1950, generations of workers in management, economics, finance, and engineering have been trained in the business of formulating linear models and solving them with simplex-based software. Often, the situations they model are actually nonlinear, but linear programming is appealing because of the advanced state of the software, guaranteed convergence to a global minimum, and the fact that uncertainty in the data can make complicated nonlinear models look like overkill. Nonlinear programming may make inroads as software develops, and a new class of methods known as interior-point

7.2. Simplexverfahren

(nach George B. Dantzig ~ 1950)

Grundprinzip: Systematisches Absuchen der Ecken des zulässigen Bereichs.

Dabei werden Austauschschritte durchgeführt, bei denen eine Basislösung durch eine "benachbarte" ersetzt wird, wobei der Wert der Zielfunktion fällt.

x : Basislösung mit Indexmenge J_B

$J = (j_1, \dots, j_n)$ Permutation von $(1, \dots, n)$ mit

$$J_B = \{j_1, \dots, j_m\}, \quad J_N = \{j_{m+1}, \dots, j_n\}$$

$$A_B := (a^{j_1} \dots a^{j_m}), \quad A_N := (a^{j_{m+1}} \dots a^{j_n})$$

$$x_B := (x_{j_1} \dots x_{j_m})^T, \quad x_N := (x_{j_{m+1}} \dots x_{j_n})^T = 0$$

Für einen beliebigen Vektor $\bar{x} \in \mathbb{R}^n$ gilt:

$$A\bar{x} = A_B \bar{x}_B + A_N \bar{x}_N = b$$

$$\Longleftrightarrow \bar{x}_B = A_B^{-1}(b - A_N \bar{x}_N) = x_B - A_B^{-1} A_N \bar{x}_N .$$

Man setze nun:

$$\boxed{D := \left(d^{j_{m+1}} \dots d^{j_n}\right) := A_B^{-1} A_N \in \mathbb{R}^{(m,n-m)}} \quad (7.9)$$

D heißt Tableaumatrix. Nach Obigem folgt damit $\bar{x}_B = x_B - D\bar{x}_N$, also

$$\begin{aligned} f(\bar{x}) &= c^T \bar{x} = c_B^T \bar{x}_B + c_N^T \bar{x}_N \\ &= c_B^T x_B + (c_N^T - c_B^T D)\bar{x}_N \\ &= f(x) + t^T \bar{x}_N, \quad \text{wobei} \end{aligned} \quad (7.10)$$

$$\boxed{t := \left(t_{j_{m+1}}, \dots, t_{j_n}\right)^T := c_N - D^T c_B \in \mathbb{R}^{n-m}} \quad (7.11)$$

t heißt Vektor der reduzierten Kosten.

t_k gibt an, um wieviel sich die Zielfunktion vergrößert, wenn man x_k, $k \in J_N$, um eine Einheit vergrößert.

Folgerung (7.12)

a) $t \geq 0 \Rightarrow x$ Lösung des lin. Programms

b) $t > 0 \Rightarrow x$ eindeutig bestimmt!

Austauschschritt:

Gelte nun für ein $r = j_q \in J_N$: $t_r < 0$.
Definiere für $\delta \in \mathbb{R}$: $x(\delta) \in \mathbb{R}^n$ durch

$$\boxed{x_B(\delta) := x_B - \delta d^r, \quad x_r(\delta) := \delta, \quad x_j(\delta) := 0 \quad (j \in J_N \setminus \{r\})} \tag{7.13}$$

Dann folgt:

$$A\,x(\delta) = b, \quad c^T x(\delta) = f(x) + \delta\, t_r \tag{7.14}$$

Denn:

- $A\,x(\delta) = A_B\,(x_B - \delta d^r) + \delta\, a^r$
  $= A_B x_B + \delta a^r - \delta A_B d^r$
  $= b + \delta\,(a^r - A_B A_B^{-1} a^r) = b$
- $c^T x(\delta) = c_B^T x_B + \delta\,\underbrace{(c_r - c_B^T d^r)}_{t_r}$ ∎

Für $\delta > 0$ fällt mithin die Zielfunktion ($t_r < 0$), während die Nebenbedingung $Ax = b$ für alle $x = x(\delta)$ erfüllt ist.

Zur Zulässigkeit:

- Ist $d^r \leq 0$, so ist $x(\delta)$ für jedes $\delta > 0$ zulässig $\Longrightarrow$ $\inf_{x \in M} f(x) = -\infty$, d.h.

  Das Problem (7.1) hat keine Lösung!

- Ist $d^r \not\leq 0$, d.h. d^r hat positive Komponten, so bestimme man $p \in \{1, .., m\}$ mit

$$\boxed{\delta := \frac{x_{j_p}}{d^r_p} := \min \left\{ \frac{x_{j_i}}{d^r_i} \,\middle|\, d^r_i > 0,\ i = 1, \dots, m \right\}} \tag{7.15}$$

Dann ist $x(\delta) \in M$, $f(x(\delta)) \leq f(x)$ und $x(\delta)$ ist Basislösung zur Indexmenge

$$J_B^{neu} := \left(J_B \cup \{j_q\} \right) \setminus \{j_p\} \tag{7.16}$$

Ferner: Ist x nichtentartet, so ist $\delta > 0$. Damit fällt die Zielfunktion nach (7.11): $f(x(\delta)) < f(x)$.

Bem.: Bei entarteten Basisvektoren x kann $\delta = 0$ auftreten. Dann bleibt $f(x)$ unveränd.

Algorithmus (7.17) (Simplex Verf.)

$A \in \mathbb{R}^{(m,n)}$, $J = (j_1, .., j_n)$ Permutation von $(1, .., n)$ mit $J_B = (j_1, .., j_m)$ Basisindizes.

① $A_B := (a_{i,j_k})_{i,k=1,..,m}$

Berechne LR-Zerlegung von A_B

Falls A_B singulär : Fehler, Stopp!

② Löse $\boxed{A_B x_B = b}$, Teste: $x_B \geq 0$ (im 1. Schritt)

$c_B := (c_{j_1}, \dots, c_{j_m})^T$, $f := c_B^T x_B$

③ Löse $\boxed{A_B^T y = c_B}$;

$t_{j_k} := c_{j_k} - y^T a^{j_k}$, $k = m+1, .., n$

$\left.\right\}$ vgl. (7.9) und (7.11)

Bestimme $q \in \{m+1, .., n\}$, $r = j_q$ mit

$t_r = \min \{ t_{j_k} : k = m+1, .., n \}$

Falls $t_r \geq 0$: Lösung gefunden, Stopp!

④ Löse $\boxed{A_B d = a^r}$

Falls $d \leq 0$: Probl. hat keine Lösung, Stopp!

Bestimme $p \in \{1, \dots, m\}$, $s = j_p$ mit

$$\frac{(x_B)_p}{d_p} = \min \left\{ \frac{(x_B)_i}{d_i} \,\middle|\, i = 1, .., m,\ d_i > 0 \right\}$$

⑤ Vertausche j_p, j_q in J, gehe zu ①

Anmerkungen:

a) In x_B stehen nur die Basisvariablen der Lösung x^*; man hat also zu setzen

$$x^*_{j_i} := \begin{cases} (x_B)_i & \text{für } i = 1, \dots, m \\ 0 & \text{für } i = m+1, \dots, n \end{cases}$$

b) In jedem Austauschschritt sind drei lineare Gleichungssysteme zu lösen. Hierzu genügt eine LR-Zerlegung von A_B und Vorwärts-/Rückwärts-Substitutionen.

c) Durch spezielle Techniken kann vermieden werden, dass im Fall entarteter Basislösungen Zyklen auftreten, also nach endlich vielen Austauschschritten wieder eine frühere Indexmenge gewählt wird.

d) Die Matrix $A_B^{(k)}$ im k. Schritt unterscheidet sich von $A_B^{(k+1)}$ nur in einer Spalte. Daher lassen sich Modifikationstechniken zum Aufdatieren der LR-Zerlegung anwenden.

Beispiel: Vgl. (7.3)

	a^1	a^2	a^3	a^4	a^5	b
A	20	10	1	0	0	8000
	4	5	0	1	0	2000
	6	15	0	0	1	4500
c^T	−16	−32	0	0	0	

1. Iteration: $J = (3,4,5 \mid 1,2)$

$$A_B = I, \quad x_B = A_B^{-1} b = \begin{bmatrix} 8000 \\ 2000 \\ 4500 \end{bmatrix}$$

$$c_B = (0,0,0)^T, \quad \boxed{f = 0}$$

$$y = A_B^{-1} c_B = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\left.\begin{aligned} t_1 &= -16 - y^T \begin{bmatrix} 20 \\ 4 \\ 6 \end{bmatrix} = -16 \\ t_2 &= -32 - y^T \begin{bmatrix} 10 \\ 5 \\ 15 \end{bmatrix} = -32 \end{aligned}\right\} \quad \boxed{r = 2 = j_5}$$

$$d = A_B^{-1} a^r = \begin{bmatrix} 10 \\ 5 \\ 15 \end{bmatrix}$$

$$\left(\frac{(x_B)_i}{d_i} \right) = (800, 400, 300)^T \Rightarrow \boxed{p = 3, \; s = j_3 = 5}$$

2. Iteration: $J = (3,4,2 \mid 1,5)$

$$A_B = \begin{bmatrix} 1 & 0 & 10 \\ 0 & 1 & 5 \\ 0 & 0 & 15 \end{bmatrix}, \quad x_B = A_B^{-1} b = \begin{bmatrix} 5000 \\ 500 \\ 300 \end{bmatrix}$$

$$c_B = (0,0,-32)^T, \quad \boxed{f = -9600}$$

$$y = A_B^{-T} c_B = \left(0, 0, -\tfrac{64}{30}\right)^T$$

$$\left.\begin{aligned} t_1 &= -16 - y^T \begin{pmatrix} 20 \\ 4 \\ 6 \end{pmatrix} = -3.2 \\ t_5 &= 0 - y^T \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \frac{64}{30} \end{aligned}\right\} \quad \boxed{r = 1 = j_4}$$

$$d = A_B^{-1} \begin{pmatrix} 20 \\ 4 \\ 6 \end{pmatrix} = \begin{pmatrix} 16 \\ 2 \\ 0.4 \end{pmatrix}$$

$$\left(\frac{(x_B)_i}{d_i}\right) = (312.5, 250, 750)^T : \quad \boxed{p = 2, \; s = j_2 = 4}$$

3. Iteration: $J = (3,1,2 \mid 4\,5)$

$$A_B = \begin{bmatrix} 1 & 20 & 10 \\ 0 & 4 & 5 \\ 0 & 6 & 15 \end{bmatrix}, \quad x_B = A_B^{-1} b = \begin{bmatrix} 1000 \\ 250 \\ 200 \end{bmatrix}$$

$$c_B = (0,-16,-32)^T, \quad \boxed{f = -10400}$$

$$y = A_B^{-T} c_B = (0, -1.6, -1.6)^T$$

$$\left.\begin{aligned} t_4 &= 0 - y^T \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = 1.6 \\ t_5 &= 0 - y^T \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = 1.6 \end{aligned}\right\} \text{ Lösung gefunden!}$$

$$\Longrightarrow \quad x^* = (250, 200, 1000, 0, 0)^T$$

$$f(x^*) = -10400$$

Satz (7.18)

Wird das Simplexverfahren (7.14) mit einer Basislösung gestartet und sind alle erzeugten Basislösungen nicht entartet, so liefert der Algorithmus nach höchstens $\binom{n}{m}$ Austauschschritten eine Lösung oder die Info., dass das Problem keine Lösung besitzt.

Behandlung freier Variablen

Ist x_j frei, d.h. $x_j \geq 0$ ist nicht vorgeschrieben, so setze man $x_j = x_j^+ - x_j^-$ mit $x_j^+, x_j^- \geq 0$.

Auffinden einer zulässigen Basislösung

Häufig ist eine zulässige Basislösung offensichtlich (wenn A die kanonischen Einheitsvektoren $e_1, .., e_m$ enthält und $b \geq 0$ gilt).

Ist dies nicht der Fall, so sorge man durch ev. Multiplikation einzelner Gleichungen mit (-1) dafür, dass $b \geq 0$ gilt und löse das folgende Hilfsproblem (Phase I des Simplexverfahrens):

> Minimiere $\quad f_H(x,y) := \sum_{i=1}^{m} y_i$
>
> unter den Nebenbedingungen
>
> $$(A \mid I_m)\begin{pmatrix} x \\ y \end{pmatrix} = b,$$
>
> $$x \geq 0, \quad y \geq 0$$

(7.19)

Offenbar ist $(x,y) = (0,b)$ eine Basislösung mit Indexmenge $J_B = \{n+1, \dots, n+m\}$.

Da die Zielfunktion f_H für alle zulässigen (x,y) nach unten beschränkt ist (durch 0), hat (7.16) eine optimale Lösung, die sich mit dem Simplexverfahren berechnen lässt.

Ist $f_H(x^*, y^*) > 0$, so hat das Ausgangsproblem keine Lösung.

Ist dagegen $f_H(x^*, y^*) = 0$, so ist auch $y^* = 0$ und x^* ist Basislösung bzgl. einer geeigneten Indexmenge $J_B \supset I(x^*)$.
